<MSc Degree Thesis>

AY 2018

# NEURAL NETWORK APPLICATION IN PREDICTING STOCK RETURNS: EVIDENCE FROM JAPAN

57161517-4 YIWEN WANG
SEMINAR ON PORTFOLIO MANAGEMENT
C.E. PROF. HITOSHI TAKEHARA
D.E. PROF. YASUHIRO ARIKAWA D.E. PROF. WILLIAM CHEUNG

## Summary

The main purpose of this paper is to forecast the one-month forward stock return by employing neural networks. The theoretical foundation and feasibility of this research are the efficient market hypothesis, which states that under weak-form efficiency, technical analysis based on the market pattern recognition is not able to predict the market movement in the future. However, we believe that traditional technical analysis or single technical indicator is not able to fully digest the current market information and hence the confirmation of market efficiency under this method is inconclusive. To solve this concern, we would like to collaborate deep learning method with technical analysis because deep learning system with nonlinear regression process is believed to recognize the market in a more comprehensive way. Our opinion upon Japanese stock market is that it might be predictable and deep learning can enhance the predicting power of technical analysis.

While similar studies have been conducted in different financial markets with various techniques, our study uses H2O platform and R language to investigate Japanese market. The combination of H2O and R is flexible and user-friendly, therefore multiple models with different characteristics are tested easily. In terms of deep learning model, multi-layer artificial neural networks with back-propagation algorithm are employed to learn historical market information, etc. price and trading volume, during learning period and test the accuracy of results during testing period.

The input data for neural networks consists of 15 technical analysis indicators and 5 liquidity

measures of individual stock, of which the total amount is 217 for our research. The input dataset is expected to predict the one-month forward return of individual stock in the back-testing period from 2010 to 2017. During the eight-year period, moving window prediction method is applied to achieve higher accuracy by using six years as the learning period all the time to process each prediction, and therefore there are two-year length of predictions in the end. The primary content after generating prediction is focused on the goodness of neural networks with different number of layers and neurons in each layer and how precise return predictions are. By comparing forecast results and actual returns, we find out models with relatively good performances and conclude that complex models with more hidden layers do not outperform simple ones. In addition, we observe advantages of neural networks over traditional linear regression using same dataset. It implies that with higher sophistication, deep learning models have greater predictive power and can be implemented for return forecast.

To conclude, there is a certain accuracy in predicting returns for Japanese stocks when using technical analysis and liquidity measures data. Further portfolio construction strategy even implies that neural networks predictions can distinguish the good stocks and bad stocks successfully, hence can provide insights for further research.

<Inside Cover>

# NEURAL NETWORK APPLICATION IN PREDICTING STOCK RETURNS: EVIDENCE FROM JAPAN

57161517-4 YIWEN WANG

SEMINAR ON PORTFOLIO MANAGEMENT

C.E. PROF. HITOSHI TAKEHARA

D.E. PROF. YASUHIRO ARIKAWA D.E. PROF. WILLIAM CHEUNG

Table of Contents

# 1. INTRODUCTION

One of main achievements of artificial intelligence techniques, the deep learning neural networks, have drawn high attention and become a popular research topic during recent decades, especially in financial discipline. Neural networks are designed to mimic human brains with neural systems and learn the past information in a higher intelligence than traditional mathematical methods. It is also able to evolve the model itself over time by constantly obtaining available messages. Due to these advanced functions, neural networks are expected to have capability of predicting stock prices and returns by many experts.

When it comes to predicting financial market, the efficient market hypothesis is the most relevant and vital topic. If the market is able to be predicted, especially by using technical analysis indicators, the market is regarded to have lower efficiency than the weak-form efficiency. In other word, when the market follows random movements, it is categorized into the higher efficient markets, the semi-efficient market or strong-efficient market. In this paper, we are interested in the weak-form efficiency, the one that is implying the least efficiency among the three. We would like to investigate the Japanese stock market particularly, which is often regarded as a developed market and hard to predict. However, with the development of new techniques, many studies have provided good example of utilizing deep learning models to generate prediction system. If our prediction models can forecast stock returns with a certain accuracy, the weak form could be criticized in Japanese stock market.

This research is conducted using H2O platform, a developed neural network tool. The technical indicators and liquidity measures are collected as inputs from year 2010 to 2017, ranging 8 years after the 2008 financial crisis. The output is one-month-forward individual stock return. Thus, our study is back testing the past and compares the actual returns with predicted ones for the same period. The input data is divided into two groups, which are learning dataset and testing dataset. A special design of this research is that moving window prediction system is applied. This moving window prediction uses 6-year learning period to forecast only one data point, which is the one-month-forward return. By moving learning periods month by month, there are in total 24 forecasted returns from January 2016 to December 2017. We are also interested in characteristics of neural networks. With different settings in terms of model characteristics, we compare the goodness of different deep learning models and discover the optimal one with specific layers and neurons number. The goodness of models is calculated by errors between predicted results and actual returns. Therefore, different models are comparable due to the same data sample and evaluation criteria. As a result, we observe that neural network models always outperform the linear regression when it comes to predict stock returns. The characteristics of neural networks is the key to generate higher accuracy so we focus on the research this topic. We find complex models with more layers or more neurons are not better off. On the contrary, the accuracy declines with the increase of layers, consistent with related studies.

The remainder of the paper is organized as follows. Section 2 briefly reviews the relevant studies concerning efficient market hypothesis, technical analysis, liquidity measures and neural network applications. The data screening for neural networks and details of model settings are described and explained in Section 3. Section 4 summaries the empirical results and provides robust test using portfolio construction strategy. Section 5 concludes our findings and gives suggestions on further work.

## 2. LITERATURE REVIEW

The efficient market hypothesis (EMH) is a well-known theory in empirical and academic finance, constantly contributed by many professionals back to early 1900s and finally refined by Fama (1970). Started from that time, the EMH has been wildly accepted with its three forms, weak form, semi-strong form and strong form. In weak form efficiency, the stock price is believed to follow a submartingale and cannot be predicted by using its past prices and trading volumes. In other words, technical analysis, which helps to find patterns of stock price movement and hence predict future trends, is not able to lead to an abnormal return. However, fundamental analysis using companies' fundamental information to evaluate firms is capable of finding undervalued opportunities and provide excess returns. In semi-strong efficiency, the stock price has already considered all of the current available information therefore even fundamental analysis cannot generate abnormal returns. The winner under this hypothesis is who have inside information that is not accessible for other market participants. The test of his hypothesis is renamed as event studies later and directly reveal the relation between the price movement and the new announcement. In strong-form efficiency, no one is able to get abnormal return because all prices have reflected all information. The last hypothesis known as the test of private information is therefore the easiest one to refute in reality.

Technical analysis is a methodology of studying past market data, using many chart tools. Dow theory states that prices trend directionally, up, down, flat or some combinations. Its followers, or technical analysts, believe in the presence of these price trends and predictability of the future direction. They also suggest the market as well as investors are repeating themselves over time, hence the market is predictable. Friesen et al. (2009) present that there is bias in acquiring and interpreting market information for traders. They find the existence of autocorrelation of price movements, regarded as the subsequent effect of trader bias, and successfully predict price jumps for some stocks. Caporin et al. (2013) contribute to technical analysis by showing that the high and low prices of equity shares are largely predictable only on the basis of their past realizations. Sezer et al. (2017) transfer the most commonly preferred technical analysis indicators into a series of buy-sell-hold trigger signals and conclude that the neural network model can achieve comparable predictions of trigger signals against the buy and hold strategy in most of the cases. In fact, majority of past related papers take a method of direction prediction, which is not aimed to forecast expected return but only buy, sell or hold signals. It is quite reasonable because technical analysis itself is playing the role of finding market turnover in the changeable environment. Usually this method will also lead to a relatively high prediction power.

Liquidity is generally described as the ability of financial securities to trade within a certain period without affecting market price. Nowadays, more and more researchers are emphasizing that liquidity plays an important role in the field of asset pricing, because as they propose, investors require higher compensation for bearing more liquidity risk. Therefore, more illiquid stocks usually have higher expected returns. Amihud (2002) estimates liquidity from the perspective of asset return and trading quantity. Liu (2006) emphasizes trading speed from the viewpoint of zero daily trading volume. Datar, Naik, and Radcliffe (1998) use the turnover ratio as a proxy for liquidity. Pastor and Stambaugh (2003) consider asset liquidity to be sensitive to innovations in aggregate liquidity from the perspective of price impact. Lesmond et al. (1999) focus on the marginal trading cost by estimating the incidence of zero returns. In brief, there are many empirical studies concerning the measure of liquidity of stocks and for this paper, the positive relationship between illiquidity and return is considered as great importance when it comes to predict stock returns.

Many markets have been found at least following the weak from EMH, especially in developed countries. Emerging markets, on the under hand, are discussed more under this topic due to high level of inefficiency and backward financial systems. Shynkevich (2012) constructs portfolio under technical strategies with U.S. small cap equities. The portfolio fails to generate abnormal returns, indicating the technical analysis has weak predictive power. Kim and Shamsuddin (2008) investigate the Asian markets and they find the Hong Kong, Japanese, Korean and Taiwanese markets are efficient in the weak-form while Indonesia, Malaysia and Philippines markets are not. The Indian stock market is tested by Sunil (1996) and is implied not weak form efficient. Lim et al. (2009) examine the weak-form efficiency of Shanghai and Shenzhen Stock Exchanges and the results indicate returns from both of markets follow a random walk, revealing the weak form EMH in Chinese markets. Alexeev and Tapon (2011) test the weak form efficiency on the Toronto Stock Exchange, using technical analysis and model-based bootstrap to generate price pattern. It has been concluded that although overall market cannot reject the weak form EMH, some sectors are less efficient than others. In addition, some scholars consider the efficiency of market can change over time. According to Cheung et al. (2011), the Hong Kong market was not weak form efficient before 1986 but may be after 1986 corresponding to the statistical test.

Although many papers have found evidences that the Japanese stock market, as one of the most developed financial markets over the world, has a relatively high level of efficiency, it is worth to retest the market with advanced technologies like neural networks. Compared with traditional models, these techniques are able to find underlying inefficiency. Previous common measurements of market efficiency include the autocorrelation-based test and the variance ratio for random walk, but since the advent of machine learning techniques, dozens of papers in recent years start to focus on return prediction with more sophisticated deep learning models. Cao et al. (2005) compare the predictive power of linear models and neural network models in Chinese stock market, showing that neural networks fit well with the emerging market and outperform the linear models.

In terms of application of neural networks, Mizuno et al. (1998) build a neural network model for stock index prediction, using some typical technical indices as inputs. It concludes that in predicting TOPIX, the neural networks outperform the single use of the technical indicator but underperform the buy-and-hold strategy. Abe and Nakayama (2018) apply deep learning models for individual stock from MSCI Japan Index. They prepare a list of fundamental factors as inputs and find the most accurate deep learning model with specific hidden layers and neurons. Shynkevich et al. (2017), who focus on S&P 500 components and use technical indicators as model inputs, observe the highest prediction accuracy when the input window length is roughly equal to the forecast window length. In brief, individual stock share is deemed to be the more reasonable research object of machine learning than stock indices. The input features and the evaluation of model predictive power are two keys for relevant research and the best design is still under discussion.

# 3. DATA AND METHODOLOGY

This section introduces the data formation, the contents of deep learning model and main body of regression methodology. The first part explains the principle of a typical knid of neural network, the multi-layer feedforward artificial neural network (ANN), followed by the second part that lists the related coding message of deep learning function in H2O platform. The dataset for back-testing and prediction modeling is described in the third part. The forth section mainly presents the detail of the input factors including 15 technical analysis indicators and 5 liquidity measures. At last, the moving-window regression method is explained in the fifth part.

## 3.1 MULTI-LAYER FEEDFORWARD ARTIFICIAL NEURAL NETWORK (ANN)

Deep learning is a crucial part of machine learning that is beneficial from the development of computer clusters recent the last decade. One of the well-established technique systems of deep learning is neural networks, which are biologically inspired by human brain and neuron structure. The simplest type of neural networks, ANN, is the main model of this paper and conducted with H2O platform and R environment.

The multi-layer feedforward ANN is also known as deep neural network or multi-layer perceptron. It consists of input layer, hidden layers and out layer and the data digestion go through all of them to get final results. Within each layer, there is a list of nodes or neurons executing designate calculations. Between layers, various algorithms can be applied to adjust the weights of layer inputs and optimize the results for different aims. A general method for nonlinear optimization called gradient descent is often implemented and also internalized with H2O. The term feedforward implies the only one data processing direction inside ANN models, usually compared with the backward feeding concept in recurrent neural network (RNN). The powerfulness of ANN is its capability of storing experiential knowledge in learning process and implementing multiple nonlinear regressions.

Another essential characteristic of neural networks is the back-propagation learning algorithm, which employs a backward phrase to minimize estimate errors after the training procedure. In RNN especially, back-propagation is the most important and complicated part of its algorithm. In this paper, we would like take advantages of H2O programming where back-propagation is natively added into its multi-layer feedforward ANN. In this paper, all mentioned deep learning models or neural networks are conducted under multi-layer feedforward ANN of H2O with consistent characteristics, introduced comprehensively in the next part.

### 3.2 APPLICATION PROGRAM H2O

H2O is a well-known and easy-to-use open resource to conduct machine learning analytics. It is chosen other than Karas and Tensorflow, which are also popular platforms for machine learning programming nowadays, because H2O is not written in Python and has a package directly bond with R. Moreover, H2O has a deep learning function based on a multi-layer feedforward artificial neural network that is trained with stochastic gradient descent using back-propagation. Therefore, all deep learning models of this paper are built within H2O deep learning function.

Though the H2O deep learning provides lots of convenience and simplifies mathematical process for beginning learners, there are some important arguments in need of manual settings. Firstly, the activation function is fixed for all models at "TanhWithDropout". Tanh activation is usually preferred in pursuit of higher accuracy to the default activation, Rectifier. At the same time, Rectifier activation has vanished gradient for negative inputs while Tanh activation has both positive and negative scales, which is preferred in return prediction of this paper. Tanh activation with dropout rate is chosen in order into reduce overfitting problems, according to Kochura et al. (2017), and the number is set at 0.1. Secondly, having a similar role with dropout, L1 and L2 penalty are applied instead of using the default. The setting is simply adding the non-zero number for L1 and L2 parameters. Thirdly, the validation set is specified to help tune the deep learning model. The validation frame can be used to stop the model earlier when overwrite_with_best_model = T and keep the optimized model without running too much rounds. An additional setting of seed is needed to generate robust results when there is a validation set. The number of seed must be the same for splitting validation set at the beginning and processing deep learning later. At last, the reproducible argument is needed to be true to make sure there is only one unique series of output. Other unmentioned arguments are not specified and using default.

Besides the above settings for all models, two special arguments, the input x and hidden, are various for different models and the accuracy of prediction could highly rely on these two arguments. The number of hidden layers and layer size are crucial for deep learning models, while there is no best combination that can be widely accepted. According to different situations, there are could be some optimization of models after trials. Therefore, 6 types of hidden layers and 2 type of layer size are designed in this paper to compare deep learning models and find out the one with relatively best performance. The number of layers follows a sequence from 1 to 6, while in each layer the number of neurons is set as either 50 or 100, corresponding to research blogs on the Internet. The neuron number is relatively big because using the available inputs to predict stock returns seems rather complex and may require more complex model with more hidden layers and neurons. Bigger size and larger number of layers might result in the overfitting problem, hence arguments like dropout is necessary. On the other hand, since we have two types of inputs, there are three alternatives for input-dependent models, which are 15 technical indicators only, 5 liquidity measures only and 20 variables together.

By running lay-dependent deep learning models, two optimal ones with either 50 neurons or 100 neurons are generated in the first part of Section 4. After generating the optimal models in both cases, the input-dependent models are conducted and to find out if there is a significant difference among different inputs.

### 3.3 DATA OF JAPANESE STOCKS

Referring to past studies that investigate different stock markets by individual stock, this paper also uses market index components and try to predict the one-month forward return of representative stocks. In terms of Japanese market, the two most indicative and leading stock indices are TOPIX and Nikkei 225. The latter one is selected for several reasons. At first, since the main inputs are technical indicators, Nikkei 225 components that are top 225 blue-chip companies are likely to have more meaningful number of historical prices and trading volumes. In other words, the effectiveness of data is one of our concerns. Moreover, this stocks universe has a more workable size. In order to cover more alternatives of inputs and maintain an acceptable running speed, the stock amount we would like to use is therefore sacrificed to some extent.

Year 2010 to 2017 is decided to be the entire horizon for this research, considering the 2008 financial crisis and the afterwards effect in Japanese market. Since the Nikkei 225 constituents are review annually, we screen all the stocks that were never be excluded in the Nikkei 225 over this 8-year period and the data sample reaches 217 stocks in the end. The industry distribution of our data sample is shown in Figure-A1 in Appendix. Input data collection includes 15 technical indicators and 5 liquidity measures of each stock. The former type is directly downloaded from Bloomberg and the latter type are calculated with historical trading volumes and prices. Involving liquidity measures differs this paper from previous studies. As a new tentative type of variables for stock return prediction, we believe it can bring great contributions since many empirical works have improved traditional asset pricing models by adding liquidity. Also, as the response or output variable of deep learning models, the monthly stock returns from 2010 to 2017 are prepared.

## 3.4 Technical Indicators and Liquidity Measures

Technical analysis is a way to predict future through the study of past, simply speaking. Its direct relation with weak form EMH is one of reasons for choosing technical analysis indicators as main part of input variables. At the same time, fundamental factors like price to earning ratio and so on are exclude in this paper in order to concentrate on one hypothesis test. From our point of view, the way technical analysis works is similar with that of neural networks. Both of tools aim to find current or future market patterns from the past, though some external drivers are not necessarily exclusive for neural networks. We want to combine and enhance the relation between technical analysis and neural network, with expectation that they can together provide good forecast of financial market.

There are 15 types of technical analysis indicators selected and displayed as follows. All of them are historical prices related and use month as period for all calculations when it is needed.

1) Percent Bandwidth (%B)

It is an indicator derived from the standard Bollinger Bands (BB) indicator. Bollinger Bands are a volatility indicator which creates a band of three lines which are plotted in relation to a security's price. The middle line is typically a 20-period simple moving average. The upper and lower bands are 2 standard deviations above and below the SMA (middle line). What the %B indicator does is quantify or display where last price is in relation to the bands. %B can be useful in identifying trends and trading signals. A buy signal is confirmed with %B bigger than 1 and a sell signal is confirmed with negative %B.

$$\%B = (last - lowerBB)/(upperBB - lowerBB), \tag{1}$$

2) Commodity Channel Index (CMCI)

It is originally introduced by Donald Lambert in 1980, only for commodities. Nowadays CCI is used for equity too, calculated as the difference between the typical price and its 20-period simple moving average ($SMA$) divided by the mean absolute deviation ($MD$) of the typical price. The typical price $p_t$ equals to the average number of the intraday highest, lowest and close prices. The inverse of 0.015 is the scale factor. The cross down +100 of CCI is considered as a sell signal and cross over -100 of CCI is considered as a buy signal.

$$CCI = \frac{1}{0.015}\frac{p_t - SMA(p_t)}{MD(p_t)}, \tag{2}$$

3) Average Directional Movement Index (ADX)

It is referred to determine the relative strength of a trend, either upwards or downwards. To calculate ADX, firstly the directional movement indicators including plus directional movement (+DM) and minus directional movement (-DM) are needed, which are featured by Wilder in his 1978 book. +DM equals current minus the prior high and -DM is prior low minus the current low, both positive. Next, divide the 14-period smoothed Plus Directional Movement (+DM) by the 14-period smoothed True Range (TR), the calculation of which is revealed in the 8$^{th}$ indicator ATR, to get the 14-period Plus Directional Indicator (+DI14). -DI14 is calculated in the same way and both of them are multiplied by 100. Then the directional movement index (DX) is computed as follows. ADX is simply a 14-day average of DX. As an indicator, when ADX is bigger than 20 it implies a trending market, while smaller than 20 implies non-trend.

$$DX = \frac{ABS[+DI14 - (-DI14)]}{+DI14 + (-DI14)} \times 100, \tag{3}$$

4) Moving Average Convergence/Divergence (MACD)

The actual input is the MACD histogram, developed by Thomaas Aspray in 1986. Prior to the introduction of MACD histogram formula, exponential moving averages (EMAs) are computed which overweight more on recent prices relative to simple moving average. An example formula for a current 10-period EMA shows as follows, where $EMA_{t-1}$ initially equals to previous 10-period SMA and $Multiplier$ equals to $2/\text{Time periods}\ +1$. The MACD histogram in fact measures the distance between MACD* and its 9-period EMA, or the signal line. That MACD* is based on 12-period EMA, the fast EMA, and 26-period EMA, the slow EMA. The bigger positive (or negative) histogram number indicates a larger divergence to the upside (or downside) trend while the smaller absolute number indicates the convergence. Simply speaking, the convergence hints the reversal of the market trend.

$$EMA_t = (Close\ -\ EMA_{t-1}) \times Multiplier +\ EMA_{t-1}, \tag{4}$$

$$MACD\ hist. = (EMA12\ -\ EMA26) - EMA9\ of\ MACD, \tag{5}$$

5) Relative Strength Index (RSI)

It is developed by Wilder as well. As a momentum oscillator, it is considered overbought when above 70 and over sold when below 30, ranging between 0 to 100. Additionally, when this signal breaks the 70 and 30 lines, it is considered divergence but has two types, bearish and bullish. For example, above 70 numbers that are declining indicate a bearish trend, or a less upside momentum.

$$Avg.\,Gain\ or\ Lo = ((previous\ 14\ day\ avg.\,gain\ or\ loss) \times 13$$

$$+current\ gain\ or\ loss)/14, \tag{6}$$

$$RSI = 100 - \frac{100}{1 + Avg.\,Gain/Avg.\,Loss}, \tag{7}$$

6) Stochastic oscillator %K (TAS_K)

Stochastic oscillators are also momentum indicators involving the previous high and low in the setting period. The default is 14 periods, which can be days, weeks or months, but here 10 periods is set for the look-back length. %K is the most volatile version of stochastic oscillator. Usually, %K above 75 is regarded as a sell signal and below 25 is regarded as a buy signal.

$$\%K = \frac{Close - Lowest\ Low}{Highest\ High - Lowest\ Low} \times 100, \tag{8}$$

7) Stochastic oscillator %D (TAS_D)

It smooths %K with a 5-day SMA, while 3-day is often the default. When %K cross above %D, a buy signal is found and vice versa.

$$\%D = 5\ day\ SMA\ of\ \%K, \tag{9}$$

8) Average True Range (ATR)

As partially mentioned in the third indicator ADX, ATR is a measure of volatility, defined by Wilder. TR is defined as he greatest of the following three, current high less the current low, current high less the previous close with the absolute value and current low less the previous close with the absolute value. At the starting point, the first 14-day prior ATR and the average of the TR values.

$$Current\ ATR = \frac{[(Prior\ ATR\ \times 13) + current\ TR]}{14}, \tag{10}$$

9) Parabolic Studies (PTPS)

This indicator is known as "stop and reverse (SAR)". For the rising SAR, it is calculated by prior SAR, extreme point (EP), which is the highest high of the current uptrend, and an acceleration factor (AF), starting at 0.02 and increasing 0.02 each time the extreme point reaches a new high. The maximum of AF is 0.2. The prior SAR for the first SAR calculation equals to the previous low. SAR is considered as a trend following indicator. Once the rising SAR cross over the price, a sell signal emerges and the falling SAR below price is also a reversal signal for buying.

$$Rising\ SAR = prior\ SAR + prior\ AF\ \times (prior\ EP - prior\ SAR), \tag{11}$$

$$Falling\ SAR = prior\ SAR - prior\ AF\ \times (prior\ SAR - prior\ EP), \tag{12}$$

10) Fear and Greed (FG)

The Bloomberg's FG indicator is the spread of two weighted 5-period moving averages of the True Range (TR). It is calculated in a way that it oscillates on a zero-based line. When FG is positive, there is a market panic and sell signal and vice versa.

11) Williams %R

Developed by Larry Williams, %R is a momentum indicator that is the inverse of fast stochastic oscillators. The setting for %R of this paper follows the default, 14 periods. The most recent close, the highest high and lowest low over the 14 periods are used in formula. The centerline is -50 in this case, where the indicator between 0 and -50 points out a buy recommendation and that between -50 and -

100 is a sell recommendation.

$$\%R = \frac{highest\ high - close}{highest\ high - lowest\ low} \times -100, \tag{13}$$

12) Momentum

It is simply the difference of current close price and previous close price. Here the previous close is set at 10-period ago.

$$momentum = \ close - close\ 10\ periods\ ago, \tag{14}$$

13) Rate of Change (ROC)

It measures the percent change in price from one period to the next. In other words, ROC compares the current price with the price some periods ago. It is also regarded as the purest form of momentum. When ROC goes down and price is trending up, it is a buy signal; when ROC goes up and price is trending down, it is a sell signal.

$$ROC = \frac{close - close\ 30\ periods\ ago}{close\ 30\ periods\ ago} \times 100, \tag{15}$$

14) Hurst Exponent

It is very special among all of indicators because a rescaled range function $R(n)/S(n)$ and time series $n$ are used. As an index of long-range dependence, if H is over 0.5, it implies a long-term positive autocorrelation, or high persistence of current trend, while the other side of 0.5 implies a reversal of trend.

$$E\left[\frac{R(n)}{S(n)}\right] = Cn^H\ as\ n \to \ \infty, \tag{16}$$

15) MaxMin Retracement

In terms of MaxMin process, there is a term called $movement$ coincide with the main trend of price and the other term called $correction$, opposite to the $movement$. The former one is the difference between the new high and previous low, while the latter one is the difference between the new high and high low. Therefore, the retracement of MaxMin of some period can be calculated as follows.

$$Retrace. = \frac{correction}{movement}, \tag{17}$$

In fact, all selected technical indicators above are not involved with stock trading volume. Considering this pitfall of data downloaded from Bloomberg, 5 liquidity proxies, ILLIQ, LOT, LMx, Turnover and Gamma are computed and comprise the other input list. Liquidity role has been highly regarded as important in recent years because it drags empirical research of stock returns closer to real market, by considering real cases like transaction costs. Many previous papers have payed attention to measure the liquidity of each stock and then update their asset pricing models. There are many ways to calculate the liquidity and this paper choose five of them with most known value of research. Like many advanced technical analysis indicators, these liquidity measures are expected to reveal some patterns after summarizing the past data.

Liquidity is not an observable variable in general. For example, many proxies for liquidity, such as the bid-ask spread, are based on microstructure data and are not available over a long-time horizon. Thus, these 5 liquidity proxies are actually computed from the angel of illiquidity rather than liquidity and cover many dimensions of illiquidity.

1) Amihud (2002) *ILLIQ*

This illiquidity measure is defined by Acharya and Pedersen in 2005, where $r_t$ is the daily stock return on day *t* and $Volume_t$ is the dollar trading volume on day *t*. This indicator can capture the price response to one dollar of trading volume, also belonging to price impact measurement.

$$ILLIQ = \sum_{t=1}^{t} \frac{|r_t|}{Volume_t}, \tag{18}$$

2) Lesmond, Ogden and Trezcinka (1999) Marginal Cost for Trades (*LOT*)

This estimator of effective transaction costs developed by Lesmond et al. in 1999. They suggest that stock true unobserved return is a function of market return, similar with CAPM model, and also assume that the marginal informed investors are rational and that they will trade only when the excess return of stock *j* above the market return exceeds transaction costs. The parameters $\alpha_{2j}$, standing for buyer's transaction costs and $\alpha_{1j}$, standing for seller's transaction costs, can be both obtained by maximizing the logarithm of likelihood function of the relation between the unobserved stock return, observed stock return and market return.

$$LOT = \alpha_{2j} - \alpha_{1j}, \tag{19}$$

3) Liu (2006) Zero-Trading Volume Day Measure (*LMx*).

It is a turnover-adjusted zero-return measure of illiquidity. $N_z$ is the number of zero trading volume days, $TVx$ is the turnover rate in the previous *x* months, $Nx$ is the number of trading days in the previous *x* months, and $DF$ is a deflator to ensure that the second term in the square brackets falls in the range of zero to one (not inclusive) for all sample stocks. In brief, this measure captures the multidimensional nature of liquidity.

$$LMx_{i,t} = \left[ N_z + \frac{\frac{1}{TVx}}{DF} \right] \times \frac{21x}{Nx}, \tag{20}$$

4) Turnover Ratio (*Turn*)

Developed by Datar in 1998, this monthly turnover is computed using daily trading shares and issued shares one year before.

$$Turn_t = \frac{number\ of\ shares\ traded\ in\ day\ t}{number\ of\ shares\ issued}, \tag{21}$$

5) Pastor and Stambaugh (2003) Return Reversal Measure ($\gamma$)

*Gamma* is also a price impact measurement like LOT, calculated by the following regression by Pastor and Stambaugh in 2003. $r_t^e$ is the security's excess return above the market return on day *t* and $Volume_t$ is the trading volume in dollars on day *t*. The coefficient on the signed trading volume, γ, is expected to be negative.

$$r_{t+1}^e = \theta + \phi r_t + \gamma sign(r_t^e)(Volume_t) + \varepsilon_t, \tag{22}$$

Since all liquidity measures are originally computed in daily basis, monthly end data of technical indicators are extracted and compose input set. The descriptive statistics of input variables are displayed in Table-A1 in Appendix, and the correlation matrix of all input variables is shown in Figure-1. There are high positive correlations between several technical indicators, such as 0.98 between Bollinger Bands (BB) and Commodity Channel Index (CMCI). It can also be observed that there is little correlation between technical indicators and liquidity data.

Figure-1: Correlation of Input Variables

Notes: The first five variables display 5 liquidity measures and the rest are technical indicators, all in abbreviated form.

| | ILLIQ | LMx | Turn | Gamma | LOT | BB | CMCI | ADX | MACD | RSI | K% | D% | ATR | PTPS | FG | %R | Mom. | ROC | Hurst | Retra. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ILLIQ | 1 | 0.05 | -0.02 | -0.04 | 0.39 | -0.06 | -0.06 | -0.09 | 0.01 | -0.12 | -0.06 | -0.08 | -0.12 | -0.14 | -0.05 | -0.07 | -0.04 | -0.12 | -0.01 | -0.14 |
| LMx | | 1 | -0.62 | 0 | -0.2 | 0.03 | 0.04 | 0.04 | -0.05 | 0.07 | 0.07 | 0.07 | -0.1 | 0.03 | 0.02 | 0.07 | 0.02 | -0.03 | 0.03 | 0.02 |
| Turn | | | 1 | 0.01 | 0.41 | -0.03 | -0.03 | -0.03 | 0.05 | -0.07 | -0.07 | -0.07 | 0.06 | -0.04 | -0.02 | -0.07 | -0.03 | 0.01 | -0.01 | -0.04 |
| Gamma | | | | 1 | -0.01 | 0 | -0.01 | 0 | -0.01 | 0.01 | 0 | -0.01 | 0.01 | 0.01 | 0 | 0 | 0 | 0.01 | 0 | 0.01 |
| LOT | | | | | 1 | 0.04 | 0.04 | 0.03 | 0.1 | -0.06 | 0 | -0.02 | -0.08 | -0.13 | 0.01 | 0 | 0.03 | -0.08 | 0.02 | -0.12 |
| BB | | | | | | 1 | 0.98 | 0.14 | 0.44 | 0.88 | 0.85 | 0.8 | -0.01 | -0.03 | 0.53 | 0.92 | 0.5 | 0.48 | 0.02 | 0.01 |
| CMCI | | | | | | | 1 | 0.13 | 0.44 | 0.86 | 0.81 | 0.81 | -0.01 | -0.03 | 0.53 | 0.87 | 0.5 | 0.48 | 0.02 | 0.01 |
| ADX | | | | | | | | 1 | 0.04 | 0.31 | 0.08 | 0.14 | 0.13 | 0.16 | 0.19 | 0.11 | 0.15 | 0.44 | 0.04 | 0.13 |
| MACD | | | | | | | | | 1 | 0.34 | 0.37 | 0.45 | -0.13 | -0.14 | 0.74 | 0.41 | 0.83 | 0.13 | 0 | -0.08 |
| RSI | | | | | | | | | | 1 | 0.76 | 0.77 | 0.06 | 0.07 | 0.55 | 0.82 | 0.51 | 0.72 | 0.03 | 0.08 |
| K% | | | | | | | | | | | 1 | 0.77 | -0.03 | -0.06 | 0.45 | 0.94 | 0.46 | 0.39 | 0.01 | 0 |
| D% | | | | | | | | | | | | 1 | -0.01 | -0.03 | 0.51 | 0.84 | 0.51 | 0.47 | 0.01 | 0.02 |
| ATR | | | | | | | | | | | | | 1 | 0.93 | 0.14 | -0.02 | 0.11 | 0.09 | -0.01 | 0.95 |
| PTPS | | | | | | | | | | | | | | 1 | 0.14 | -0.05 | 0.11 | 0.11 | -0.01 | 0.96 |
| FG | | | | | | | | | | | | | | | 1 | 0.5 | 0.83 | 0.34 | 0 | 0.2 |
| %R | | | | | | | | | | | | | | | | 1 | 0.49 | 0.43 | 0.01 | 0.01 |
| Mom. | | | | | | | | | | | | | | | | | 1 | 0.32 | 0 | 0.18 |
| ROC | | | | | | | | | | | | | | | | | | 1 | 0.04 | 0.07 |
| Hurst | | | | | | | | | | | | | | | | | | | 1 | -0.01 |
| Retra. | | | | | | | | | | | | | | | | | | | | 1 |

## 3.5 DYNAMIC TRAINING WINDOW

A common way to run deep learning model is to split the sample data to training, validation and test sets. When the sample data is fixed, the prediction numbers of test period will come out at one time after training and validation. Abe and Nakayama (2018) suggest to use a moving window for training and testing to predict the one-month forward return for each stock. In their paper, the latest N month value of input variables comprise a set of training data to predict the stock return, while most of input variables are fundamental factors. In addition, Skabar and Cloete (2002) apply the similar methodology while using technical indicators as input variables. Both of papers claim that by dynamic training and testing data over time, the prediction model can achieve better optimization because it is allowed to change from past to future. Therefore, referring to this methodology, this paper also uses the moving window for training and validation. In the 6-year training and validation procedure, neural networks analyze the relation between 20 input variables and one-month forward returns for individual stock. After optimizing the model, return of the first month after that 6-year period is predicted based upon testing-period input dataset, which is across only one month. For instance, the first deep learning model of stock *i* will use its technical indicators and liquidity measures from January 2010 through

November 2015, as the set of inputs. The corresponding output is the return series of that stock from February 2010 to December 2015. After accomplishing the first deep learning model, its return of January 2016 is predicted using input data of December 2015 and the next model is made in the same way by both moving the training period and testing period one month forward. In the end, the stock *i* will have 24 values of predicted returns from January 2016 to December 2017.

The same sets of data are regressed linearly for return prediction, as the counterpart of neural networks. Similarly, a set of 6-year input variables are explanatory factors, which are defined as $X_j$ at time *t* in formula below and the return in one month is the explained factor. The whole regression will move month by month to predict return series.

$$Ret_{t+1} = \alpha_t + \beta_{jt} X_{jt} + \mu_t, \tag{23}$$

All stocks are regressed under this moving-window process separately. The result of deep learning regressions and linear regressions can be comparable due to the same regression dataset.

# 4. EMPIRICAL RESULTS

In this section, deep learning models and linear models are compared for their goodness under many situations. In the first part, the predictive neural network models with different settings of layer and layer size are contrasted. The results align with our expectation that the complex model is not necessarily better than simple models with few hidden layers. In the second part, the input is adjusted for the multi-layer models with relatively optimized performances from the first comparison to show how the embedded information varies for different type of inputs. A simple portfolio construction strategy is applied in the last part to evaluate the prediction power of neural networks.

## 4.1 PREDICTION RESULTS OF MULTI-LAYER MODELS

We build one neural network model for every one-month forward return, hence from January 2016 to December 2017, we have 24-month or 5208 predicted returns for all stocks derived from 12 deep learning regression models and one linear model respectively. Since the non-linear models share the exact same data sample with the linear model, the result as well as the goodness of model are comparable.

The common evaluation metrics available for regression models includes Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). There is no perfect evaluation method currently for deep learning models. Therefore, by defining as follows, this paper chooses simply MAE and RMSE as evaluation metrics for neural networks.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{24}$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{25}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{26}$$

**Table-1: MAE and RMSE of deep learning models**

Notes: Evaluation metrics MAE and RMSE are computed by the real-time returns and predicted returns. The column shows the results with different model settings, etc. 1 hidden layer with 50 neurons inside for the first cell. The last row shows the goodness of the linear regression model, using all inputs as explanatory variables for one-month-forward return.

| | 50 neur. | | 100 neur. | |
|---|---|---|---|---|
| Hidden layer(s) | MAE | RMSE | MAE | RMSE |
| 1 | 5.0005 | 6.5451 | 4.4308 | 5.8721 |
| 2 | 4.9184 | 6.5078 | **4.2180** | **5.6266** |
| 3 | 4.8271 | 6.4628 | 4.2917 | 5.7330 |
| 4 | **4.7369** | **6.3715** | 4.4176 | 5.9559 |
| 5 | 5.1255 | 6.8640 | 4.7850 | 6.4469 |
| 6 | 5.3215 | 7.1462 | 5.0918 | 6.8311 |
| Linear | 5.3799 | 7.2284 | 5.3799 | 7.2284 |

*All number is in percentage

MAE measures the average magnitude of the errors in a set of predictions, without considering their directions. It is the average over the test sample of the absolute differences between predictions and actual observations where all individual differences have equal weight. On the other hand, RMSE is the square root of the average of squared differences between predictions and actual observations. It is believed that RMSE has the benefit of penalizing large errors more than MAE because the errors are squared before they are averaged. Therefore, RMSE gives a relatively high weight to large errors. Both methods indicate that the lower the number, the better the model is.

The goodness of models with different neuron numbers and layer numbers is summarized in Table-1. It is found that in general models with100 neurons in each hidden layer have smaller MAE and RMSE than those having 50 neurons in each hidden layer, indicating more neurons lift the accuracy level. Another important observation is that a good model is not necessarily to have as many layers as possible, consistent with our expectation. For example, given all models with 50 neurons in each hidden layer, the best model with lowest MAE and RMSE has four hidden layers, and given all models with 100 neurons in each hidden layer, the best model with lowest MAE and RMSE has only two hidden layers. Therefore, it also implies that there is no general optimal solution for neural networks. By involving in the results of linear regression model, a conclusion can be drawn that all deep learning models outperform the simple linear regression. In all cases, the MAE and RMSE of linear regression is the upper bound for others.

The Figure-2 displays the relation between the number of hidden layer and evaluation metrics for models with 50 neurons, while the Figure-3 is made for 100-neuron models. There is no big difference between the trend of MAE and RMSE corresponding to hidden layers and both curves are U-sharp with optimal solutions. As a result, we find if the deep learning model with 4 hidden layers and 50 neurons in each performs the best among all 50-neuron models. The model with 2 hidden layers and 100 neurons in each even better than previous one, with the lowest MAE and RMSE.

**Figure-2: Deep learning models with 50 neurons in each hidden layer**

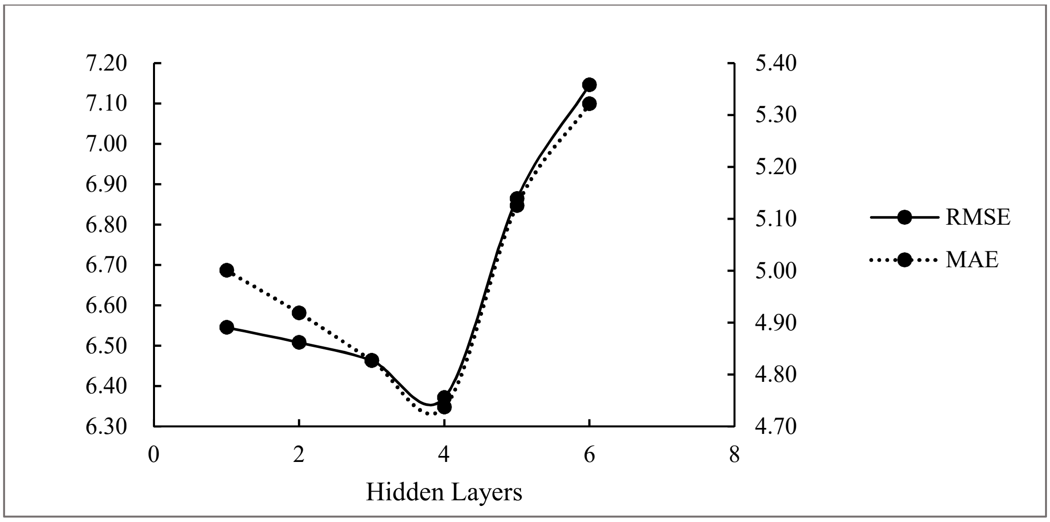
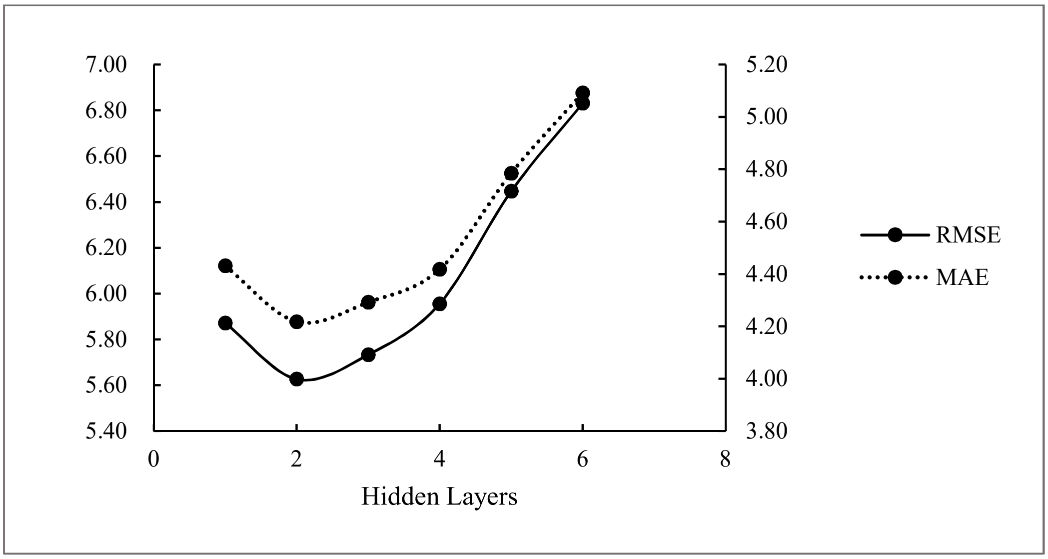



**Figure-3: Deep learning models with 100 neurons in each hidden layer**



## 4.2 PREDICTION RESULTS OF INPUT MODELS

Since best models with 50 and 100 neurons respectively have been observed, further comparisons are shown in Table-2 depending on changes of inputs.

The four hidden-layer model with 50 neurons in each layer and two hidden-layer model with 100 neurons in each layer are selected from the previous work. At the same time, the linear model using same sets of data is conducted, showing its goodness in Table-2. A new evaluation method is introduced to compare models with different types of input in this section, called Volatility of Forecasted Errors (VFE). It is the standard deviation of forecast errors. The VFE and RMSE have almost the same values but the VFE is more direct and easier to understand intuitively. When it comes to models with different size of samples, especially for the comparison in this part, the VFE is preferred for more precise estimation of models.

It can be concluded that both technical indicators and liquidity measures contribute to the deep learning prediction because models with all inputs have the lowest VFE compared with technical-indicator-only models and liquidity-measure-only models seen in Table-2. Technical indicators might be better input for return predictions than liquidity measures because technical-indicator-only models have much lower VFE than liquidity-measure-only models. The reason why technical indicators surpass liquidity measures are various. One possible answer could be that there are 15 technical indicators and only 5 liquidity measures, thus the limitation of liquidity measures may result from the lack of computation methods. Technical indicators, on the other hand, are more developed in past decades.

**Table-2: VFE of neural networks and linear regression**

Notes: Volatility of Forecasted Errors (VFE) is the standard deviation of forecast errors and it is used to evaluate models having different input scales. The object of comparison includes two models with the lowest RMSE and MAE having 50 and 100 neurons respectively.

| | 4*50 | 2*100 | Linear |
|---|---|---|---|
| All Inputs | 6.3695 | 5.6265 | 7.1976 |
| Tech. Only | 6.8165 | 5.9534 | 6.7223 |
| Liquid. Only | 8.1689 | 8.0394 | 8.9180 |

## 4.3 PORTFOLIO CONSTRUCTION

Based upon the above comparisons, it is hard to draw further conclusions concerning the goodness of neuron network models. After all, the deep learning process is hidden in so called "black box" and evaluation metrics such as MAE and RMSE tell limited information about the goodness of models. Therefore, a portfolio construction strategy is employed to help understand the capability and application of predicted results.

The best model assessed previously with lowest errors is the one with 2 hidden layers and 100 neurons in each, hence predictions of this model is implemented as the representative of deep learning forecasting. Before moving to further statistic test, it has been revealed by some example histograms, e.g. Figure-A2 and Figur-A3 in Appendix that predicted returns of each month follow a roughly normal distribution, so the student *t* test can be applied to test the statistical significance.

The methodology of portfolio construction strategy that is to build equally weighted high-return portfolios and low-return portfolios under neural networks and linear regression respectively. At first, 217 of return predictions in each predicting month are sorted in descending order. The top-50 return group and bottom-50 return group are formed 24 times during the two-year period. The next step is to equally weight each stock in top and bottom categories, generating High-return Portfolio (HRP) and Low-return Portfolio (LRP) in each month. The stock composition could be different over time, indicating both of portfolios need to be monthly rebalanced. At last, the return of portfolios is calculated based on the actual return of each composition and the series of return spreads between high and low is tested with student t-test. In the following paragraph, HRP-DP and LRP-DP refers to portfolios based on the best deep learning model and HRP-LR and LRP-LR indicates portfolios constructed with linear regression. All predictions are generated with all types of inputs.

HRP-DP shows average 7.50% monthly return and LPR-DP shows average monthly return of -4.61%, seen in Table-3. Meanwhile, the average monthly return of Nikkei from 2016 to 2017 is 1.24%, as the benchmark. Rreturn of HRP-LR is 6.67% and return of LRP-LR is -4.15%. The difference between top portfolio and bottom portfolio is large and statistically significant, since both *p-value* are smaller than 5%.

**Table-3: Results of Welch's two-sample *t* test**

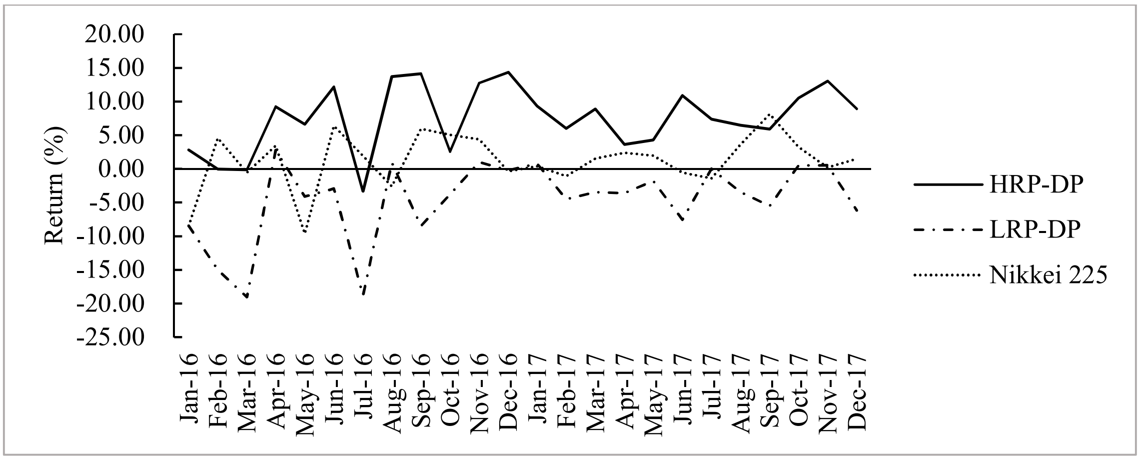
Notes: In statistics, Welch's *t* test, or unequal variances t-test, is a two-sample location test which is used to test the null hypothesis that two populations have equal means. Welch's *t* test is an adaptation of Student's *t* test and is more reliable when the two samples have unequal variances and unequal sample sizes. Here 95% confidence interval is used.

| | | | |
|---|---|---|---|
| Mean (%): HRP-DP | 7.5023 | Mean (%): HRP-LR | 6.6689 |
| Mean (%): Nikkei 225 | 1.2434 | Mean (%): Nikkei 225 | 1.2434 |
| Diff. (%) | 6.2588 | Diff. (%) | 10.8215 |
| *t* statistic | 4.7684 | *t* statistic | 4.1845 |
| *p-value (95% C.I.)* | 0.0000 | *p-value (95% C.I.)* | 0.0000 |

Portfolios built under deep learning regression have bigger *t* statistics, indicating that deep learning method distinguish the well-performed and poor-performed stocks better than simple linear regression. As long as the good and bad stock shares are apart, it is easier for investors to apply further strategies, like long-short strategy, and generate excess return.

As a result, the return of high-return portfolios predicted by deep learning models precede the benchmark Nikkei 225 index and low-return portfolios both with great statistical significance. The larger number of *t* statistic in terms of the difference between HRP-DP and Nikkei 225 than that in terms of the difference between HRP-LR and Nikkei 225 shows that the deep learning model has greater forecasting power than linear regression. We draw return series of HRP-DP, LRP-DP and Nikkei 225 from 2016 to 2017 in Figure-4. Return curve of HRP-DP is always above that of LRP-DP and monthly returns of Nikkei 225 are roughly between the two curves. The trend of all curves is similar and shows co-movement in general.

**Figure-4: Return of HRP-DP, LRP-DP and Nikkei 225**



## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we explore the prediction function and relative performance of artificial neural networks with H2O platform. The hidden layers and layer size of neural networks attach a great importance for prediction power and for particular problem like return prediction in this paper, specific scales of hidden layers and neurons are pre-set to find the optimal combination. As for input, this paper chooses technical indicators and liquidity measures to predict stock return in one month with moving windows, which help improve the robustness of model. The main conclusion is that neural network model with all of inputs and 100 neurons in 2 hidden layers has the best forecast results than others, depending on the evaluation metrics MAE and RMSE, and both technical and liquidity proxies make contribution to a better prediction. The further application of neural network predictions is to build ranked portfolios. It has been found out that relatively high return stocks predicted by neural networks have significantly higher actual return than that of relatively low return stocks grouped in the same way, hence the prediction lead by neural networks is able to provide a certain of accuracy in predicting the Japanese stock market.

Based on the current results and achievement, there are further progresses can be made. The linear regression already built in this research is based on weak-form efficiency hypothesis, therefore only market data are used as input. To check out higher efficiency, some fundamental factors like asset size and book-to-market ratio can be used to build another linear regression and see if the result is better than the deep learning model. In terms of tools, there is a more advanced function of H2O called H2O grid search that is capable of optimizing the hyperparameter, etc. hidden layer(s) and neurons. If this function is conducted, there could be a more precise model of neural networks with not only 50 or 100 neurons. This function is quite time-consuming, while the run time needed is already an existing problem in current research where each stock needs at least twenty minutes to complete one prediction. The trade-off between running time and prediction accuracy has been a primary research topic in deep learning field, hence there could be more research of different in our case to find a better balance point in case of expanding our stock scale to the larger universe.

# REFERENCES


[1] Abe, M. and Nakayama, H. (2018), “Deep learning for forecasting stock returns in the cross-section,” *In PAKDD 2018, Advances in Knowledge Discovery and Data Mining*, 273-284.

[2] Acharya, V. and L. H. Pederson (2005), “Asset pricing with liquidity risk,” *Journal of Financial Economics*, 77 (2), 475-410.

[3] Alexeev, V. and Tapon, F. (2011), “Testing weak form efficiency on the Toronto Stock Exchange,” *Journal of Empirical Finance*, 18 (4), 661-91.

[4] Amihud, Y. (2002), “Illiquidity and stock returns: Cross-section and time-series effects,” *Journal of Financial Markets*, 5 (1), 31-56.

[5] Cao, Q., Leggio, K.B. and Schniederjans, M.J. (2005), “A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market,” *Computers and Operations Research*, 32 (10), 2499-2512.

[6] Caporin, M., Ranaldo, A. and Santucci de Magistris, P. (2013), “On the predictability of stock prices: A case for high and low prices”, Journal of Banking and Finance, 37 (12), 5132-5146.

[7] Cheung, W., Lam, K.S.K. and Yeung, H. (2011), “Intertemporal profitability and the stability of technical analysis: evidences from the Hong Kong stock exchange,” *Applied Economics*, 43 (15), 1945-63.

[8] Datar, V.T., N. Y. Naik and R. Radcliffe (1998), “Liquidity and stock returns: An alternative test,” *Journal of Financial Markets*, 1 (2), 203-209.

[9] Fama, E. (1970). “Efficient Capital Markets: A Review of Theory and Empirical Work”. *Journal of Finance*, 25 (2), 383–417.

[10] Friesen, G.C., Weller, P.A. and Dunham, L.M. (2009), “Price trends and patterns in technical analysis: A theoretical and empirical examination”, *Journal of Banking and Finance*, 33 (6), 1089-1100.

[11] Kim, J.H. and Shamsuddin, A. (2008), “Are Asian stock markets efficient? Evidence from new multiple variance ratio tests,” *Journal of Empirical Finance*, 15 (3), 518-32.

[12] Kochura, Y., Stirenko, S. and Gordienko, Y. (2017), “Comparative performance analysis of neural networks architectures on H2O platform for various activation functions,” *In 2017 IEEE International, Young Scientists Forum on Applied Physics and Engineering (YSF)*, 70-73.

[13] Lesmond, D.A., J. P. Ogden and C. A. Trzcinka (1999), “A new estimate of transaction costs,” *Review of Financial Studies*, 12 (5), 1113-1141.

[14] Lim, K., Habibullah, M.S. and Hinich, M.J. (2009), “The Weak-form Efficiency of Chinese Stock Markets: Thin Trading, Nonlinearity and Episodic Serial Dependencies,” *Journal of Emerging Market Finance*, 8 (2), 133-63.

[15] Liu, W. (2006), “A liquidity-augmented capital asset pricing model,” *Journal of Financial Economics*, 82 (3), 631-671.

[16] Mizuno, H., Kosaka, M., Yajima, H. and Komoda, N. (1998), “Application of neural network to technical analysis of stock market prediction,” *Studies in Information and Control*, 7 (3), 111-120.

[17] Pastor, L. and R. F. Stambaugh (2003), “Liquidity risk and expected stock returns,” *Journal of Political Economy*, 111 (3), 642-685.

[18] Poshakwale S. (1996), “Evidence on weak form efficiency and day of the week effect in the Indian stock market,” *Finance India*, 10 (3), 605-616.

[19] Sezer, O., Ozbayoglu, A. and Dogdu, E. (2017), “A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters,” *Procedia Computer Science*, 114, 473–480.

[20] Shynkevich, A. (2012), “Performance of technical analysis in growth and small cap segments of the US equity market,” *Journal of Banking and Finance*, 36 (1), 193-208.

[21] Shynkevich, Y., T. M. McGinnity, S. A. Coleman, A. Belatreche and Y. Li (2017), “Forecasting price movements using technical indicators: Investigating the impact of varying input window length,” *Neurocomputing*, 264, 71–88.

[22] Skabar, A. and Cloete, I. (2002), "Neural networks, financial trading and the efficient markets hypothesis," *Australian Computer Science Communications*, 24 (1), 241-249.

# APPENDIX

**Figure-A1: Histogram of Stock Industry Distribution**

Notes: Our dataset is comprised of 217 stocks that belong to Nikkei 225 from 2010 through 2017. The Industry distribution of our stock universe show as follows.

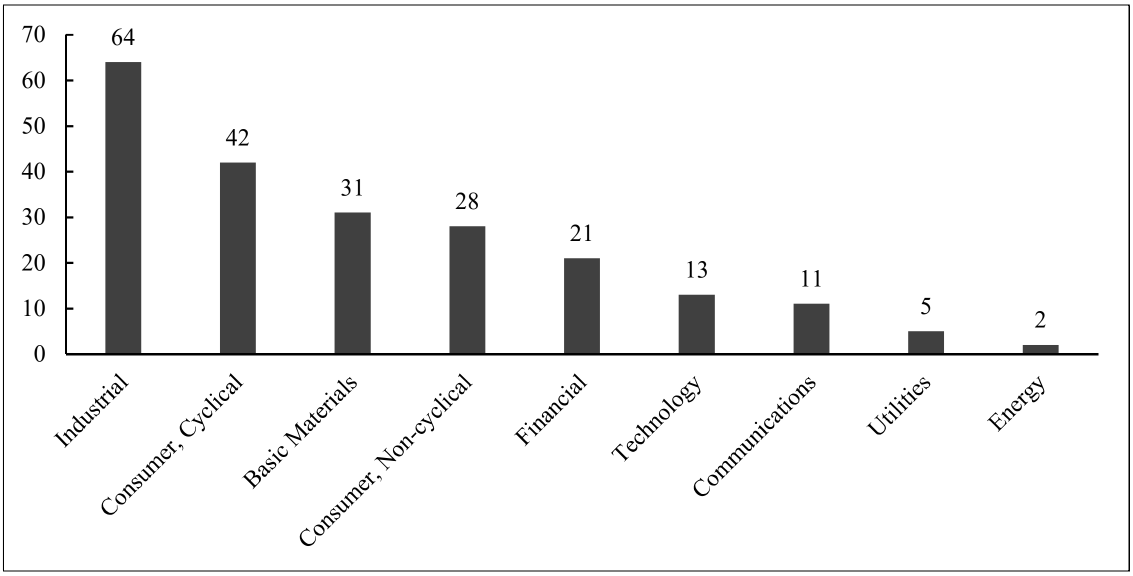



**Table-A1: Descriptive Statistics of Model Variables**

Notes: Several descriptive statistics of one response variable (Ret) and twenty input variables are displayed in this table. The return (Ret) number is in percentage and show in the first column. The descriptive statistics include the average (Mean), median, mode, standard deviation (Std. D), variance (Var.), minimum and maximum of time series data from 2010 to 2017.

| | Ret | ILLIQ | LMx | Turn | Gamma | LOT |
|---|---|---|---|---|---|---|
| Mean | 1.2262 | 0.1234 | 0.0001 | 71.2599 | 0.0000 | 0.0077 |
| Median | 1.0519 | 0.0624 | 0.0001 | 55.1515 | 0.0000 | 0.0062 |
| Mode | 0.0000 | 0.0272 | 0.0001 | 35.2943 | 0.0000 | 0.0063 |
| Std. D | 8.9152 | 0.2438 | 0.0000 | 63.7079 | 0.0002 | 0.0075 |
| Min | -77.9565 | 0.0014 | 0.0000 | 10.5583 | -0.0042 | 0.0000 |
| Max | 76.1628 | 3.3641 | 0.0004 | 926.3438 | 0.0046 | 0.0695 |

**Table-A1: Descriptive Statistics of Model Variables (continued)**

| | B% | CMCI | ADX | MACD | RSI | TAS_K |
|---|---|---|---|---|---|---|
| Mean | 1.3786 | 28.6 | 24.4 | 18.0 | 53.0 | 54.3 |
| Median | 1.4727 | 32.8 | 22.9 | 13.2 | 51.8 | 55.9 |
| Mode | 0.5000 | -57.7 | 17.4 | - | 39.1 | 50.0 |
| Std. D | 3.2097 | 107.8 | 9.2 | 105.8 | 11.6 | 28.2 |
| Min | -9.3655 | -366.8 | 7.4 | -3,072.0 | 17.7 | 0.0 |
| Max | 10.4183 | 437.5 | 65.1 | 1,827.3 | 93.4 | 100.0 |

**Table-A1: Descriptive Statistics of Input Variables (continued)**

| | TAS_D | ATR | PTPS | FG | WLPR | MOMENTUM |
|---|---|---|---|---|---|---|
| Mean | 54.0 | 301.8 | 2,231.2 | 149.2 | -45.3 | 138.5 |
| Median | 55.3 | 218.2 | 1,614.8 | 66.8 | -43.4 | 62.8 |
| Mode | 52.9 | 195.0 | 3,040.0 | 187.4 | 0.0 | -20.0 |
| Std. D | 23.5 | 403.1 | 2,921.9 | 959.2 | 28.2 | 1,007.4 |
| Min | 2.3 | 13.9 | 86.8 | -19,167.7 | -100.0 | -31,485.0 |
| Max | 98.2 | 7,730.7 | 61,970.0 | 25,319.0 | 0.0 | 24,665.0 |

**Table-A1: Descriptive Statistics of Input Variables (continued)**

| | ROC | HURST | RETRACEMENT |
|---|---|---|---|
| Mean | 23.3 | 0.7110 | 2,226.5 |
| Median | 8.7 | 0.6173 | 1,629.8 |
| Mode | 0.0 | 0.4367 | 1,252.0 |
| Std. D | 67.0 | 3.6371 | 2,794.4 |
| Min | -94.9 | -44.3243 | 131.5 |
| Max | 674.4 | 161.4512 | 46,460.0 |

**Figure-A2: Histogram of Oct. 2017 Predicted Returns by Deep Learning**

Notes: This histogram depicts return predictions of 217 stocks in October 2017, from the two-layer neural network model that has 100 neurons in each layer with all inputs. Horizontal axis captures equal interval of monthly returns from -14.4% to 21.6% and vertical axis counts the number of returns in each interval.

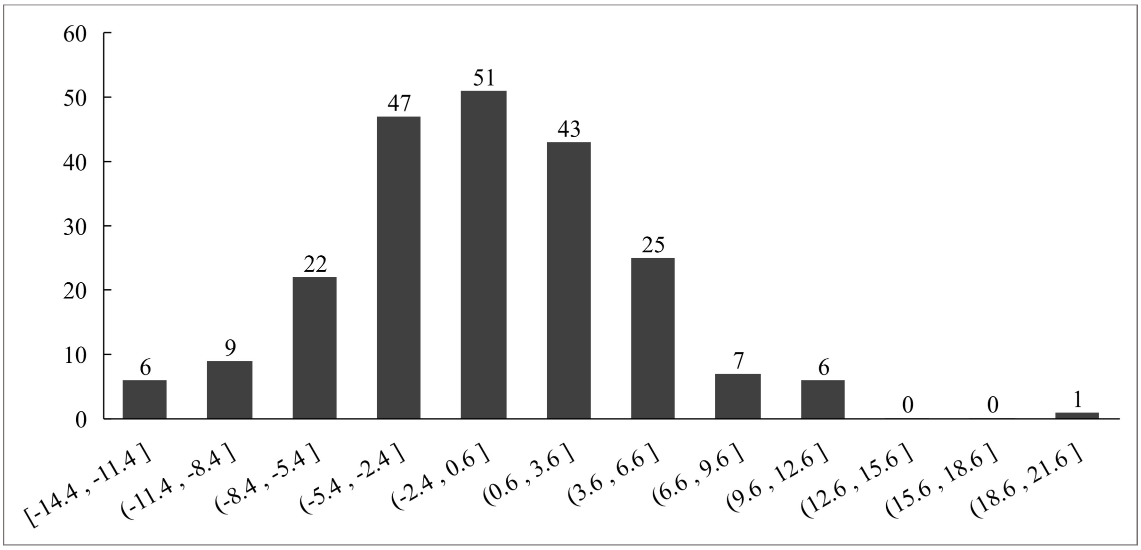



**Figure-A3: Histogram of Oct. 2017 Predicted Returns by Linear Regression**

Notes: This histogram depicts return predictions of 217 stocks in October 2017, from the linear regression model with all inputs. Horizontal axis captures equal interval of monthly returns from -14.4% to 21.6% and vertical axis counts the number of returns in each interval.